DOCUMENT RESUME

ED 317 164 IR 014 233

AUTHOR Wardle, Andrew; O'Connor, Rodric
TITLE MATS--Management Accounting Tutorial System. Version 1.0. Project Documentation.
INSTITUTION Manchester Univ. (England).
PUB DATE 89
NOTE 23p.; For related reports and a user guide, see IR 014 231-234 and IR 014 253-255.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Accounting; *Business Education; *Computer Assisted Instruction; Computer System Design; *Databases; Foreign Countries; Higher Education; Instructional Development
IDENTIFIERS University of Manchester (England)

ABSTRACT

The Management Accounting Tutorial System (MATS) is a management accounting database for a carpet manufacturing company. The system allows the display and output of monthly activities, and is intended to provide a means of illustrating the main topics of the second year management accounting course at Manchester University. The system itself provides no tutoring as such, but the outputs of the system are geared to illustrate aspects of various topics. The steps in the development of MATS are described: (1) systems analysis; (2) objectives and criteria specification; (3) programming system selection; and (4) testing. A diary of project development activities and discussion of limitations of the system are included. Possibilities for future enhancement of the system (either to improve the current application or to develop MATS beyond the current application) are outlined. (GL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED317164

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

PROJECT DOCUMENTATION

MATS - Management Accounting Tutorial System

Version 1.0

Andrew Wardle **Rodric O'Connor**

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Jean Shaoul
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

IR014233

TABLE OF CONTENTS

1.1) INTRODUCTION		2
1.2) TERMS OF REFERENCE		2
2.1) SYSTEMS ANALYSIS		3
2.2) DESIGN OBJECTIVES		4
2.3) CHOICE OF PROGRAMMING SYSTEM		5
2.4) DIARY OF PROJECT DEVELOPMENT		8
3) TESTING		13
4.1) CURRENT LIMITATIONS		16
4.2) FUTURE DEVELOPMENT		19
APPENDICES		
I	SYSTEMS LAYOUT	22
II	FILE STRUCTURES	27
III	SCREEN LAYOUTS	35
IV	PROGRAM LISTINGS	43
V	TESTING OUTPUT	129

1.1) INTRODUCTION

MATS is a management accounting database for a carpet manufacturing company. The system allows the display and output of the monthly activities, and is intended to provide a means of illustrating the main topics of the second year management accounting course. The system itself provides no tutoring as such; the outputs of the system are however geared to illustrate aspects of the various topics.

1.2) TERMS OF REFERENCE

The idea for the MATS stemmed from the desire to develop teaching of the Manchester University second year undergraduate management accounting course using a simple illustrative company. The proposal was originally made by **Mr Peter Taylor** and **Doctor Jean Shaoul**. KPMG Peat Marwick McLintock were approached in order to derive some basic data for the exercise.

At this point the idea was offered to the students of Computer Science and Accounting for a final year project. Originally, four students responded, ourselves and another pair. Originally intended to be a half year exercise, the other group developed a spreadsheet model and a single tutorial exercise as a half year project. We continued the development on a full year basis.

Mr Peter Taylor and Doctor Jean Shaoul acted as supervisors throughout, and **Ms Anne Kirkham** of KPGM Peat Marwick McLintock was consulted on a weekly basis during the year.

2.1) SYSTEMS ANALYSIS

Before any programming began, we undertook a study of objectives and requirements. This primarily involved informal interviews with Peter Taylor and Anne Kirkham in order to ascertain exactly what was required. It soon became apperent that the objectives of the exercise were fairly open-ended and that the design of the final system was not known. Rather the program would develop gradually and expand to meet further requirements as thay were identified.

Basic Requirements

The overall objective of the exercise was to be the presentation of management accounting information of a single company sufficient to illustrate the main management accounting topics of Machester University's second year management accounting course. The following areas were highlighted:

1) Background and history
2) Costing theory and budgets
3) Cost allocation
4) Standard costing
5) Variances
6) Decision making (with respect to pricing and product mix)
7) Sensitivity analysis
8) Cost centres and divisionalisation
9) Transfer pricing
10) Performance evaluation
11) Company structure

All of the above were to be included in a computer model of the company which could be used to pull down relevant information for illustrating the various accounting treatments for teaching purposes. The system output was to be used by the tutor to aid the student by presenting figures that were realistic, and to demonstrate the interrelationship between many of the topics. Emphasis would be on interpretation and evaluation rather than on number-crunching.

Data was provided in a series of handouts. These contained standard costing specifications for each product, together with sales, production and indirect cost levels. Budget data was provided followed at a later date by the actual figures.

It soon became clear that we faced an enormous task in incorporating all of the above areas into the system, due to limitations imposed by both our own work schedules and limited opportunities for consultations. We therefore prepared a work schedule for project development (see section 2.4) which proved to be an accurate reflection of our output. The topics in the above list were given relative priorities with no strict limitation on how much or little should be achieved.

2.2) DESIGN OBJECTIVES AND CRITERIA

In the designing of the system we had several objectives. Probably the most important was **flexibility**, so that the system could cope with the increasing requirements identified as time progressed, and also provide a basis for future development.

Since we did not know on what kind of computer sytem the programs would be run we had to ensure **portability**, so that the program would be transferable to different IBM personal computers with the minimum of modification.

We decided that programming would be on a modular basis, primarily due to the nature of the dBase environment (see 'Choice Of Programming System'), but also so that we achieved a good level of **maintainability** ; the program had to be easy to change or modify when the need arose, especially since the final objectives of the development were unknown.

We also decided to place an emphasis upon **readability**, since it became apparent that it is likely that future development will be carried out.

This was at the cost of **performance**, since some of the programming solutions were not necessarily the quickest or most efficient methods. This approach was justified when we were asked to include an absorption cost generator very similar to the existing standard costing generator some twelve weeks after the latter had been completed. The loss of performance is not a serious drawback as the major time-consuming processes are only run very rarely.

These critera were followed in the development on the management accounting tutorial system, and an adequate compromise between the different paradigms was achieved.

2.3) CHOICE OF PROGRAMMING SYSTEM

Following discussions with Anne Kirkham and Jean Shaoul we proceeded with a Lotus 123 implementation of the requirements as identified in the introductory hand-out received from Anne. A spreadsheet was set up which we felt was satisfactory, but we discovered that the Lotus model was to present restrictions both of memory and user friendliness. It is for these reasons that we decided to consider the possibility of using dBASE III to implement the system.

i) THE LOTUS MODEL

We developed a spreadsheet model with the following format:

OUTPUT P + L accounts, Balance Sheets (annual, year to date)

INPUT Entry of monthly actual figures

DATA e.g. Standard Cost Tables, monthly stock levels

CALCULATIONS of budget and year to date figures, variances etc...

MACROS -Fixed format printouts, shell macro, menus etc.

The model was set up using a COMPAQ 386/20 with a 1024 Kb RAM, and when tried on the the 512 Kb IBM PC's in the Roscoe Computer Laboratory it was discovered that there was insufficient memory space to store the entire worksheet, and it was decided that in order to achieve the objectives satisfactorily the spreadsheet would have to be split up into smaller units to be loaded into a main sheet.

The format was as follows :

Main Worksheet
Budgeted and Actual Annual Profit and Loss a/c and Balance Sheet
Budgeted and Actual y-t-d Profit and Loss a/c and Balance Sheet
Data (as above)
Macros
Area for input of monthly actual figures

Other files :
Twelve files containing monthly budgets
Twelve files containing actual monthly results (to be created by the main spreadsheet when monthly figures are entered)
files containing background info, assumptions, credits etc.

The worksheet would be driven by a menu macro which would run continually

Using this model there are limitations not immediately apparent. Firstly, the combining of files is very time consuming (Unforgivable on a 16 bit machine), and this would be exacerbated by the use of the Local Area Network in operation in the PC Lab. Also the ability to edit monthly figures would become difficult with the y-t-d figure adjustments.

PROBLEMS WITH THE LOTUS MODEL

i) Time restrictions of file accesses necessary for the system.
ii) Memory (RAM) limitations.
iii) Difficulty in implementing range and validity checks for input data.
iv) Poor user interface (especially macros).

ADVANTAGES

i) Easy implementation of sensitivity analysis etc...
Ii) Excellent graphing facilities.
iii) Sophisticated users can make minor (major?) changes if required.

SUMMARY

Basically we have to balance the need for speed, user-friendliness and simplicity with the physical limitations of both the IBM machines and the Local Area Network. Certainly, given the required hardware and a degree of user sophistication, the Lotus model would provide an ideal environment under which to implement the system.

Having experimented however, we came to the conclusion that a much more satisfactory solution could be achieved using a database management system such as dBASE III +

A justification for this conclusion follows.

ii) THE dBASE III PLUS MODEL

dBase III provides an environment in which a more user- friendly system can be created which will produce a more satisfactory implementation of the requirements. When necessary dBase III files can be down-loaded into Lotus for analysis purposes by creating a set of empty but formatted worksheets which would accept information from the dbase files. The data could then be analysed using all of the Lotus model's data manipulation and graphing facilities. Thus the advantages of the spreadsheet model are not lost with the dBase model.

Specifically :

ADVANTAGES

1) A secure, user friendly environment is easily achievable.
2) Simpler presentation and task (requirement) separation.
3) Validity checks and range checks are possible.
4) Flexibility both in editing of data files and for expansion of the system in the future.
5) Satisfactory storage method
6) 'Tailor made' report generation
7) Ability to restrict access if this becomes necessary.

DISADVANTAGES

1) Development time increases
2) Sensitivity analysis etc...
NB. Files can be downloaded into Lotus 123
3) Very limited graphics facilities are available.
4) The use of the system requires dBase III, software which is not as readily available as Lotus within the university and perhaps elsewhere. This restricts the user base although the software has been specifically developed for use by just Peter Taylor, and problems of portability could of course be solved by compilation.

2.4) DIARY OF PROJECT DEVELOPMENT

October

10th	Initial meeting. Received introductory handout (1)
12th	Discussion with Peter re system specification (2)
18th-25th	Development of Lotus prototype (32)
25th	Discussions with Anne re layout of spreadsheet Identification of Lotus constraints (2)
31st	dBase feasibility study (10)

November

1st	Discussions with Anne re stock calculation, design of dBase files (5)
8th	Discussions with Anne, verifying stock figures (1)
11th - 15th	Lotus macro development (6)
15th	Discussions with Anne (dBase feasibility) and Peter (systems analysis) (3)
17th-21st	Presentation preparation (20)
18th	Discussions with Jean re Lotus constraints and possible solutions (1)
21st	Project presentation (1) Go-ahead for dbase system
28th	dBase programming began
29th	GSALES, GCOSTS, GBFCF, GELEC, GLAB (20)
30th	GMISC, CALPUR, GENSTD (13)

December

1st	Designing of FIFO stock system (5)

2nd	OPST, NEWSTK, (6)
3rd	UPSTK, SUMSTK, CALSTK (9)
4th	Discussions with Jean re interfacing between Lotus and dBase (2)
5th	MENU.PRG (5)
6th	Received actual results data from Anne (1)
7th-9th	DISPBUD (15)
12th	Debugging of programs (10)
13th	Discussions with Anne re displaying of accounting figures (1)
15th	GYTDGE (5)
16th	DISPYTD (4)
17th	Entering of actual data (8)
21st	GETMONTH (2)
28th-31st	Extending of generator for actuals (16)
31st	DISPBUD updated to include actuals (6)

January

2nd	GETVID.BIN (3)
3rd-9th	STDVAR (30)
10th	MVARDP (8)
11th	YVARDP (4)
12th	FAVADV (3)
13th-17th	Debugging of variances (12 approx.)
17th	Discussions with Anne and manual testing of variances (8)
20th	Discussions with Peter re output requirements and print module

	prototypes (4)
22nd-27th	PSELECT (25)
24th	Discussions with Anne re design of outputs (1)
26th	Restructuring of program modules (5)
31st	Discussions with Anne re absorption costing (1)

February

5th	ABSSALES, ABSBFCF, ABSCALPU, ABSCOSTS, ABSRATE (10)
7th	Discussions with Anne, ABSELEC, ABSLAB, ABSMISC (6)
8th	ABSNEWST, ABSOPST, ABSCALST, ABSSUMST, ABSUPSTK (6)
9th	ABSYTDGE, ADJUST (6)
11th-17th	Debugging of absorption costing modules (15)
13th	DISPBUD and DISPYTD updated for absorption costing (6)
14th	Discussions with Anne re user interface design and menu options (1)
17th	PSELECT extended for absorption costing (3)
19th-21st	Variances for absorption costing (not used) (6)
20th	ASSUMP, BACKGRD, BALSHT (4)
21st	Discussions with Anne re details of context in which to present the company (1)
22nd-26th	DEF (15)
26th-5th March	PERFREP (40)
28th	Discussions with Anne re performance reports (1)

March

3rd	FAVADVP, TMENU (4)
6th	PRACT, PRACTSDC (6)
7th	Discussions with Anne re absorption costing bases (1)
8th	PRBUD, PRBUDSDC (6)
9th	Discussions with Peter re report format layouts, tutorial data requirements, interface prototype (1)
10th	Discussions with Jean re future developments and Lotus interface (1)
11th	ERROR (5)
12th-15th	Report formats created , *.FRM (20)
14th	Discussions with Anne re tax, dividends, miscellaneous items (1)
15th-17th	GENTAX (6)
19th	GENMENU, UPBAR (6)
20th	STANDARD, ABSORB (4)
21st	Discussions with Anne re tutorial formats, menu layouts and items (2)
22nd-24th	TUTMENU (5)
23rd	Discussions with Peter to demonstrate tutorial menu prototype and tutorial requirements (2)
24th	Restructuring, making programs consistent (10)
25th	TITLE, FTOH.BAT, HTOF.BAT (4)
26th	FMENU, HMENU (1)
27th	Testing - creation of a spreadsheet for this purpose (14)
28th	Discussions with Anne re testing procedures, Budget, Actual testing (14)
29th	Zero data testing (14)
30th	Stock generator testing (9)

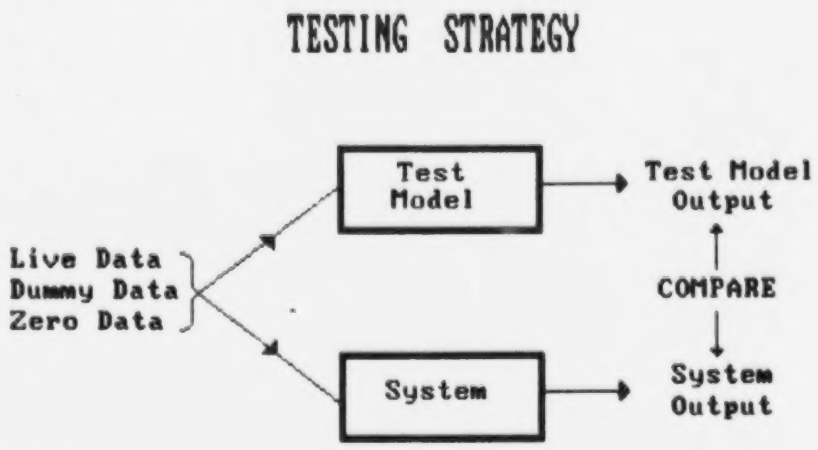
31st-2nd	Variance testing (9)

April

1st-13th	User Documentation (75)
2nd - 13th	Project Documentation (70)
2nd	ADDPATH.BAT (1)
6th	Discussions with Peter re future development over the summer and tutorial introductions (1)
7th	Discussions with Anne re documentation content, Outputs to Peter (2)
8th	CHECKPR, TESTPR (6)
13th	Finished Documentation
14th	Documentation to Anne for proof reading (10)
15th	Walkthrough with inexperienced user (3)
17th	Meeting with Anne to collect proof read documentation (1)
21st - 26th	Finished documentation (21)

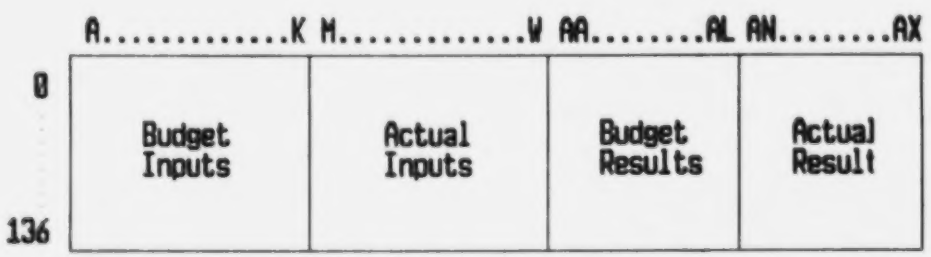
TOTAL 770 hrs (approx)

3) TESTING



We employed a **black box** approach to testing the majority of the system. This involved comparing system outputs with the results obtained from a simple model of the system. In addition, logic testing was carried out for the variance and stock calculation modules.

In order to test the system output for correctness a spreadsheet model was created to preform the major mathematical operations carried out by the system. The worksheet took the following format:



The time taken to set up the worksheet was justified as it was very flexible and enabled us to quickly run dummy data tests. Further, the model could be adapted as the Lotus extension of the MATS that has been discussed.

During testing we uncovered a number of minor errors and points of judgement which have been resolved. For example, the absorption rate was originally calculated on a monthly basis when in fact Peter wanted it to be worked out on an annual basis. Apart from a few minor problems, the success rate was very pleasing.

See Appendix V for test results

The system input data was inserted in this worksheet and the outputs of the two models compared. This process was repeated with dummy data and zero data to confirm outputs.

The use of the spreadsheet in the testing procedure was purely as an aid to calculation. It was originally intended that the process would be done manually but the sheer volume of calculations involved was prohibitive. The model, set up by Andrew, was kept as simple as possible, and all cell formulas were confirmed by Rodric in an independent check.

In addition to the main tesing procedure, many of the program modules were tested during development soon after being written. Anne also tested the correctness of the accounting formulas involved in the variance and absorption costing programs, and compared system outputs with manually developed solutions.

Due to the problems in simulating a FIFO stock system in the lotus model, we tested the **logic** of the stock generation programs. A copy of the program was modified so that the numbers used during program execution were directed to a file. This file was then examined to confirm that the stock levels and transactions corresponded to those calculated manually. Each line of the code was then examined to ensure that it was logically correct, and that the program could handle all possible types of input data.

We have carried out two 'walkthroughs' with independent people to test the efficiency of the user interface.

The first, Liam McNally, was a second year Umist student with virtually no knowledge of computers or accounting. We asked him to read the user manual, and then set a number of tasks for him to perform with the system. While he attempted to do these, we observed the particular difficulties he was having and offered no help whatsoever. The major finding of this exercise was the lack of any understanding about the keyboard and it's special keys. In addition, our input methods proved to give insufficient prompting and no feedback. For example, Liam would type '11' rather than '1' as he

knew nothing of the auto-repeat of most keyboards. Once this mistake had been made, the system offered no help as to what had gone wrong, but merely requested the input again. This type of observation has enabled us to improve the interface for the highly inexperienced user.

The walkthrough was repeated with Peter Taylor, who will be the main user of the system at present, to identify any more specialised aspects of both the interface and the presentation of the accounting information.

Further to this, the documentation has been tested and proof-read by two independent parties. The first, Miss Mara Jones, has had experience in using computer applications in her work. Anne Kirkham was very helpful in proof reading the entire first draft of the documentation, and offering suggestions about omissions and redundancies.

4.1) LIMITATIONS OF THE SYSTEM

Input Data

At present the inputs to the system are in the form of eleven database files which cannot be changed in any way within the system. This was not an initial requirement of the system although we feel it is a feature which should be added in the future to give the system more flexibility and enhance it's illustrative capabilities.

At present MATS assumes that all input data is true and good. Such range checking and quantity limitation can easily be carried out within input programs.

Variation of product lines

The system generators, for simplicity, work on the assumption that only product line A varies between budget and actual. It would be a relatively simple task to extend this to all of the four product lines.

Scope of topics

By the end of programming we had not incorporated as many topics as we had wished. This stemmed largely from the fact that the development of the dBase system only began at the end of November, following experiments with a Spreadsheet model. At present there is no distinction between departments, the costing systems are fixed and no balance sheet data is of relevance to the Profit and Loss Accounts. In addition, no pricing decisions are made andno demand considerations have been included.

Sensitivity Analysis

At present no form of sensitivity analysis is available within the system. The data files have been incorporated into Lotus however so that some analysis can be carried out on costing considerations and production levels. In addition, Lotus can then be used to produce graphical information which cannot be generated within the MATS.

Generation Time

Running on an XT on the local area network from a floppy disk the full generation program can take over an hour. We do not however see this as a serious setback as the generator modules only have to be run when the input data is changed. In addition, compilation would reduce running time by some 80%, and program refinement could reduce the number of disk accesses required in generation.

Field widths

The database storage method places restrictions upon the size of figures within the files. They can however be increased at the cost of storage and memory space. At present, most numeric fields are nine digits wide (capacity = 999,999,999)

Screen Size

In EGA mode the size of the text screen can be increased to 44 rows x 80 columns, instead of the default of 24 x 80. We have not however utilised this feature as we were unaware of the hardware we would be using.

The dbase environment

At present the system must be run within the dBase environment. This not only makes for slower execution as modules must be interpreted , but also restricts the portability of the system to computers supporting the dBase III Plus system. The problem could be solved by compilation of the program modules into a machine code program executable from the disk operating system. There are a number of dBase compilers available at present for as little as sixty pounds, and the new dBase IV has its own inbuilt compiler.

Interface

The very basic nature of the user interface is excusable at present because of the relatively simple nature of user activity, although with further development the range of input methods should be increased to include for example some form of mouse control.

Illustrative Qualities

We feel that at present the system does not illustrate how calculations are performed in arriving at outputs. This conforms to the original concepts of the design in that the emphasis is interpretive, stressing evaluation techniques rather than mathematical processes.

Help

We attempted to implement an on-line help system for the MATS. This involved developing a pascal routine to save the screen so that help could be displayed at the touch of a key. Problems were encountered however, since this involved accessing an external program which had to be loaded into memory. It became clear that the dBase environment did not release all of the memory used during such an operation, so that the help system crashed after a few runs. We thus abandoned the idea for the moment, since there will be only one user of the system and anything that is not self-explanatory is covered in detail in the User Guide. The problem could probably be overcome in the future however and the exercise served to give us a better knowledge of the problems that further development of on-line help presents.

Disk Space

Since there is an option to divert output to a file, the limitations of disk space should be considered. The size of such file outputs is however indeterminable, and we have not placed and restriction on this. If there is insufficient disk space the operation is abandoned and control returned to the main menu.

Errors

The error trapping routine writes a description of the error type and its date of occurence, but does not indicate where in the system the error occured in the error log.

Printer Requirement

The output programs, both to printer and to file, require that a printer be on line. A binary routine has been included to check printer status and restrict access to modules requiring the printer to when the printer is not on line.

4.2) FUTURE DEVELOPMENT

There are numerous possibilities for future enhancements to the management accounting tutorial system. They can be divided into two separate categories, enhancements to the system that improve the current application i.e. coverage of more tutorial (topics), and enhancements that evolve MATS beyond the current application such as single and multi user business games.

These are listed in the order that we feel they should be implemented.

1. Variation in products B,C and D

At the moment only product A varies from budget. The current generation modules will cope will varying B,C and D although variance generation only calculates variances in A. The display modules and the performance report generator would also have to be modified.

2. Input programs

As it stands the input data is taken from dBase files that have been set up using the dBase command 'BROWSE'. There is no check on the quality of the input i.e. range testing, and we have had to assume that the system input files are made up of 'true' data. Thus the input module must check that the data is valid by checking ranges and ensuring that no numeric overflows result from generation calculations.

3. Department split

This would require new method of generation to allow evaluation of departmental performance.

4. Individual Product accounts

Displaying of profitability and performance reports of each product.

5. Stock valuation methods

Incorporate different stock valuation methods.

6. Pricing decisions

Incorporating a demand function so that sales would be generated, not input to the system, based upon price, a demand function and a random element.

7. Product mix decisions

The opportunity cost of producing each product.

8. Transfer Pricing between departments

Once an inter-department relationship has been established, the use of transfer pricing techniques for manager evaluation would improve the teaching scope of the system.

9. Absorption cost basis extension

It was originally intended that figures were calculated on a number of absorption costing bases, such as direct labour hours, factory space and production units. So far only the production unit base has been made. In the present state, alternative methods would have shown very small differences.

10. Multi-company data

The design of the system is such that it will support any number of different sets of company figures at the same time. This would be achieved by making directories for each company with a full set of input and output data in each. The system could then be extended to change the current data directory easily to switch between companies.

11. Cash flow statements and balance sheets

The generation of cash flow statements and balance sheets to tie in with profit and loss accounts would add realism. This will involve cash flow assumptions

12. Previous year data

The ability to store more than one years company information, together with the ability to cope with the transition at year end would provide comparability between years. This would provide an introduction to trend analysis in the system.

13. On-line help system

The problem of memory limitations within dBase might be solved with the use of a number of flags to act as keys for a help database.

14. Actual data generation

The idea here is that the user will input budget data and the system will generate actual data based upon random variations and significant departures from standard. Thus, for example, pricing decisions could be reflected in sales. This already forms the basis of a very simple business game, which could be extended to include a wide variety of management decisions for a comprehensive business simulation.

Looking further ahead, the network provides an environment for a multi user business game. The scenario here might be that each team is responsible for a company in the carpet industry, whether supplying,manufacturing or retailing, competing and interacting with each other in a simulated environment affected by random economic fluctuations.